UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/716,916	11/18/2003	Greg Christopher JR.	07844-602001-P555	6410

21876 7590 11/28/2007

FISH & RICHARDSON P.C.
P.O. Box 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
CHEN, QING	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
11/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MN

Office Action Summary	Application No.	Applicant(s)	
	10/716,916	CHRISTOPHER, GREG	
	Examiner	Art Unit	
	Qing Chen	2191	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 September 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5-12,14-23 and 25-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3, 5-12, 14-23, and 25-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on September 4, 2007.

2. **Claims 1-3, 5-12, 14-23, and 25-32** are pending.

3. **Claims 7, 10, and 14-16** have been amended.

4. **Claims 4, 13, and 24** have been cancelled.

5. **Claims 25-32** have been added.

6. The objection to the specification is withdrawn in view of Applicant's amendments to the claims.

7. The objections to Claims 2, 3, and 5-9 are maintained in view of Applicant's arguments and further explained below.

8. The 35 U.S.C. § 112, second paragraph, rejections of Claims 7 and 14-16 are withdrawn in view of Applicant's amendments to the claims. However, the 35 U.S.C. § 112, second paragraph, rejection of Claim 3 is maintained in view of Applicant's arguments and further explained below.

9. The 35 U.S.C. § 101 rejection of Claim 24 is withdrawn in view of Applicant's cancellation of the claim.

Response to Amendment

Claim Objections

10. **Claims 2, 3, 5-9, and 30** are objected to because of the following informalities:

- **Claims 2, 3, and 5-9** recite the statutory category of invention "The method."

Applicant is advised to change this statutory category of invention to read "The machine-

implemented method" for the purpose of keeping the claim language consistent throughout the claims.

- **Claim 30** recites the limitation "the data." Applicant is advised to change this limitation to read "the installation data" for the purpose of providing it with proper explicit antecedent basis.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

11. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

12. **Claims 3 and 27** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 3 and 27 recite the limitation "about one day (emphasis added)." The term "about" is a relative term, which renders the claims indefinite. The term "about" is not defined by the claims nor does the specification provide a standard for ascertaining the requisite degree and one of ordinary skill in the art would not be able to reasonably determine the scope of the invention. In the interest of compact prosecution, the Examiner subsequently does not give any patentable weight to this limitation for the purpose of further examination.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. **Claims 1-3, 5-12, 14-18, and 25-32** are rejected under 35 U.S.C. 103(a) as being unpatentable over **US 6,738,970 (hereinafter Kruger)** in view of **US 6,560,776 (hereinafter Breggin)**.

As per **Claim 1**, Kruger discloses:

- generating a comparison of a current software installation, to a target computer, with a previous software installation, to the same target computer, in a series of two or more software installations *(see Column 7: 51-62, "Difference calculator 234 compares the tree stored in before tree storage 230 with the tree stored in after tree storage 232 to determine which changes have taken place to the master computer.")*;

- creating installation data for a resource, based at least in part on the comparison, the resource including attributes including a dynamic attribute and a static attribute, the dynamic attribute being an attribute that should have changed between the previous software installation and the current software installation, the static attribute being an attribute that should remain unchanged between the previous software installation the current software installation *(see Column 5: 18-29, "In both of the preceding embodiments, leaf nodes of the subtree or subtrees*

correspond to files, and contain information about the files in place of the files themselves. In one embodiment, such information is referred to as the node's properties and contains some or all of the file details ... These details may include the filename, last modification date, size, access permissions such as read only, and security information describing who is allowed access to the file and the type of access allowed."; Column 6: 51-58, "Registry file state retriever 225 reads the operating system registry file, such as the windows registry file in Microsoft Windows 95, and builds a subtree corresponding to the hierarchy of the registry file. For example, the Windows registry file arranges the keys and values in a hierarchical folder system and this hierarchy is used to build the subtree. Leaf nodes hold the values in the node's properties, and parents of these nodes store the keys in their properties."); and

- identifying from the installation data the dynamic attribute that was not changed in the current software installation *(see Column 8: 34-40, "When difference calculator 234 compares a terminal node, the properties of the node are also compared, and if the properties of each corresponding node are the same, difference calculator 234 marks the terminal node in the tree it creates as the "same". This means the state represented by the terminal node did not change when the new software was installed.").*

However, Kruger does not disclose:

- software product development; and

- presenting potential problems with the current software installation based on the identified dynamic attribute to facilitate verification of an installer for the software product development.

Breggin discloses:

- software product development *(see Column 3: 66 and 67 to Column 4: 1-6, "... the install program is created by a builder or installer on a computer that is hereinafter referred to as the build computer. The builder or installer writes a program or script describing how the software and supporting files are to be installed on a target computer.")*; and

- presenting potential problems with the current software installation based on the identified dynamic attribute to facilitate verification of an installer for the software product development *(see Figure 5; Column 10: 3-16, "Referring to FIG. 5, the display provides, for each exception, the file name("FILE"), the file location ("LOCATION"), the file size("SIZE"), the last modification date ("DATE"), the file version("VERSION"), and the registration status ("REG").")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Breggin into the teaching of Kruger to include software product development; and presenting potential problems with the current software installation based on the identified dynamic attribute to facilitate verification of an installer for the software product development. The modification would be obvious because one of ordinary skill in the art would be motivated to provide a user with useful diagnostic information for a software product under development.

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Kruger further discloses:

- identifying from the installation data the static attribute that was changed in the current software installation *(see Column 8: 49-56, "If the properties in the table corresponding to a terminal node of the before table are different from the corresponding table entry of the*

after node but have the same filename (for file terminal nodes) or same parent key (for value nodes), difference calculator marks the node as changed.").

However, Kruger does not disclose:

- presenting potential problems with the current software installation based on the identified static attribute to facilitate verification of the installer for the software product development.

Breggin discloses:

- presenting potential problems with the current software installation based on the identified static attribute to facilitate verification of the installer for the software product development *(see Figure 5; Column 10: 3-16, "Referring to FIG. 5, the display provides, for each exception, the file name("FILE"), the file location ("LOCATION"), the file size("SIZE"), the last modification date ("DATE"), the file version("VERSION"), and the registration status ("REG").").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Breggin into the teaching of Kruger to include presenting potential problems with the current software installation based on the identified static attribute to facilitate verification of the installer for the software product development. The modification would be obvious because one of ordinary skill in the art would be motivated to provide a user with useful diagnostic information for a software product under development.

As per **Claim 3**, the rejection of **Claim 1** is incorporated; however, Kruger does not disclose:

- wherein the previous software installation is performed about one day prior to the current software installation.

Official Notice is taken that it is old and well-known within the computing art to perform software installation on a daily basis. Applicant has submitted in the originally-file specification that the resources needed to correctly install a software application can change regularly, often on a daily basis *(see Page 1, Paragraph [0002])*. As a result, daily installation is performed to ensure that the software application is kept up-to-date with the most recent resource changes. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the previous software installation is performed about one day prior to the current software installation. The modification would be obvious because one of ordinary skill in the art would be motivated to ensure that a software application is kept up-to-date with the most recent resource changes.

As per **Claim 5**, the rejection of **Claim 1** is incorporated; however, Kruger does not disclose:

- tracking expectations for the resource in a primary installation baseline and a secondary installation baseline, and wherein presenting the potential problems comprises presenting a baseline-update interface by transmitting markup language data.

Breggin discloses:

- tracking expectations for the resource in a primary installation baseline and a secondary installation baseline, and wherein presenting the potential problems comprises presenting a baseline-update interface by transmitting markup language data *(see Column 10: 40-*

42, "In Web-based applications, the installed database or file can be incorporated into one or more web pages." and 49-67 through Column 11: 1-5, "In this process, a baseline file, which is simply a 'snapshot' of the exceptions on the target computer at a given time, is generated manually or automatically. The baseline file can be used to 'mask' or remove previous exceptions from the installed file or database." and "This feature permits a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be installed." and "... after the user has selected the baselining option ... the processor in box 240 opens and reads the baseline file(s). In box 244, the processor iteratively compares the contents of the baseline file(s) with the list of exceptions and other pertinent information in the installed database of file(s).").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Breggin into the teaching of Kruger to include tracking expectations for the resource in a primary installation baseline and a secondary installation baseline, and wherein presenting the potential problems comprises presenting a baseline-update interface by transmitting markup language data. The modification would be obvious because one of ordinary skill in the art would be motivated to permit a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be identified *(see Breggin – Column 10: 61-64).*

As per **Claim 6**, the rejection of **Claim 1** is incorporated; however, Kruger does not disclose:

- excluding a set of resources from the generated comparison for the software product development.

Breggin discloses:

- excluding a set of resources from the generated comparison for the software product development *(see Column 3: 14-15, "The exceptions can be filtered to exclude known exceptions from analysis."; Column 10: 59-61, "Using the base lining process, these exceptions can be excluded from further displays of exception data."; Column 11: 5-8, "Any matching items are removed from the list of exceptions to be displayed graphically to the user.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Breggin into the teaching of Kruger to include excluding a set of resources from the generated comparison for the software product development. The modification would be obvious because one of ordinary skill in the art would be motivated to permit a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be identified *(see Breggin – Column 10: 61-64).*

As per **Claim 7**, the rejection of **Claim 5** is incorporated; however, Kruger does not disclose:

- wherein expectations of resource changes, including the installation data, are stored in a relational database indexed by date, platform, language, and product configuration.

Official Notice is taken that it is old and well-known within the computing art to index data in a relational database using various attributes. Data in a database is often indexed by

various attributes pertaining to the particular application of the data. For example, software installation data is often indexed in a database by platform (operating system), supported languages, and product configuration information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein expectations of resource changes, including the installation data, are stored in a relational database indexed by date, platform, language, and product configuration. The modification would be obvious because one of ordinary skill in the art would be motivated to store and access additional useful data in the database pertaining to the software installation.

As per **Claim 8**, the rejection of **Claim 1** is incorporated; and Kruger further discloses:

- wherein the attributes comprising modification date stamp information, file size information, security permissions information, and checksum information *(see Column 5: 25-29, "... such information is referred to as the node's properties and contains some or all of the file details ... These details may include the filename, last modification date, size, access permissions such as read only, and security information describing who is allowed access to the file and the type of access allowed.")*.

As per **Claim 9**, the rejection of **Claim 1** is incorporated; and Kruger further discloses:

- wherein the resource comprises a file and a system registry, and the installation data comprises deletions, additions, and modifications of the resource *(see Column 5: 58-67, "The nodes corresponding to the files themselves are built as nodes, though not leaf nodes, by ini file state retriever 222. Although the nodes corresponding to files do contain the same information*

(name, file size, etc.) as the ordinary files described above, ini file state retriever 222 builds child nodes descending from the file nodes."; Column 6: 51-58, "Registry file state retriever 225 reads the operating system registry file, such as the windows registry file in Microsoft Windows 95, and builds a subtree corresponding to the hierarchy of the registry file."; Column 8: 49-67 through Column 9: 1-4, "If the properties in the table corresponding to a terminal node of the before table are different from the corresponding table entry of the after node but have the same filename (for file terminal nodes) or same parent key (for value nodes), difference calculator marks the node as changed." and "... difference calculator 234 adds a node in the tree it builds, marks that node as "deleted" ..." and "... difference calculator 234 adds the node into the tree it builds using the same lineage as the after tree, marks the node as "added" ...").

Claim 10 is a software product claim corresponding to the machine-implemented method claim above (Claim 1) and, therefore, is rejected for the same reason set forth in the rejection of Claim 1.

As per **Claim 11**, the rejection of **Claim 10** is incorporated; and Kruger further discloses:

- receiving input specifying which of the identified dynamic attribute and static attribute should be static in their installation data for future software installation *(see Column 12: 13-36, "For example, if the location of the windows directory is at c:\windows, shell processor searches for "c:\windows" in all nodes of the supertree. Shell processor replaces "c:\windows" with the alias $windows. This allows the program that will perform the installation on a*

subsequent machine to adjust the location to match the corresponding location on the subsequent machine."); and

- designating a new expectation of stability for the specified attribute according to the received input *(see Column 12: 41-46, "After post processor 250 has completed its operation, the resulting tree is referred to as manifest. Post processor 260 places the manifest in manifest storage 260. The manifest tells an installation program on any subsequent machine how to make the changes that will perform the installation on the subsequent machine.").*

Claim 12 is rejected for the same reason set forth in the rejection of Claim 2.

Claim 14 is rejected for the same reason set forth in the rejection of Claim 5.

Claim 15 is rejected for the same reason set forth in the rejection of Claim 6.

Claim 16 is rejected for the same reason set forth in the rejection of Claim 7.

Claim 17 is rejected for the same reason set forth in the rejection of Claim 8.

Claim 18 is rejected for the same reason set forth in the rejection of Claim 9.

Claims 25-32 are system claims corresponding to the machine-implemented method claims above (Claims 1-3 and 5-9) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-3 and 5-9.

15. **Claims 19, 20, 22, and 23** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Breggin** in view of **Kruger**.

As per **Claim 19**, Breggin discloses:

- a build controller *(see Figure 6: 4; Column 4: 1-6, "... the build computer.")*;

- an install controller comprising a database including a baseline recording expectations *(see Figure 1: 200; Column 7: 47-49, "... the processor places the information into the installation database or file."; Column 10: 50-52, "The baseline file can be used to 'mask' or remove previous exceptions from the installed file or database.")*; and

- wherein the build controller automatically triggers the install controller to initiate installer tests as part of a software build process, and collects test results to be presented in a report comprising a baseline-update interface *(see Figure 3B: 236; Figure 5; Column 4: 16-21, "... the (build) computer first reads in ... the installation program or script ... and creates a list of program files, data files, and/or registry entry changes ... and writes certain of this information to the installation database."; Column 9: 55-58, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format."; Column 10: 17-28, "As illustrated in by FIG. 5, exceptions can be displayed selectively at differing levels depending, for example, on the field to which the exception pertains.")*.

However, Breggin does not disclose:

- a dynamic attribute and a static attribute for one or more resources associated with a software installer, the dynamic attribute being an attribute that should have changed between a previous software installation and a current software installation, the static attribute being an attribute that should remain unchanged between the previous software installation and the current software installation;

- one or more install slave machines; and

- the install controller automatically dispatches installation to the one or more install slave machines.

Kruger discloses:

- a dynamic attribute and a static attribute for one or more resources associated with a software installer, the dynamic attribute being an attribute that should have changed between a previous software installation and a current software installation, the static attribute being an attribute that should remain unchanged between the previous software installation and the current software installation *(see Column 5: 18-29, "In both of the preceding embodiments, leaf nodes of the subtree or subtrees correspond to files, and contain information about the files in place of the files themselves. In one embodiment, such information is referred to as the node's properties and contains some or all of the file details ... These details may include the filename, last modification date, size, access permissions such as read only, and security information describing who is allowed access to the file and the type of access allowed."; Column 6: 51-58, "Registry file state retriever 225 reads the operating system registry file, such as the windows registry file in Microsoft Windows 95, and builds a subtree corresponding to the hierarchy of the registry file. For example, the Windows registry file arranges the keys and values in a hierarchical folder system and this hierarchy is used to build the subtree. Leaf nodes hold the values in the node's properties, and parents of these nodes store the keys in their properties.")*;

- one or more install slave machines *(see Column 4: 1-5, "The master computer is any computer on which the computer software can be properly installed, and for which such*

installation will be used as a model for installation of the software on other computer systems."); and

- the install controller automatically dispatches installation to the one or more install slave machines *(see Column 4: 19-27, "The systems sends the instructions, files, and program to other computer systems using conventional management software ... When the program sent is operated, it can install the computer software in a manner consistent with the manner the computer software was installed on the master computer system.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kruger into the teaching of Breggin to include a dynamic attribute and a static attribute for one or more resources associated with a software installer, the dynamic attribute being an attribute that should have changed between a previous software installation and a current software installation, the static attribute being an attribute that should remain unchanged between the previous software installation and the current software installation; one or more install slave machines; and the install controller automatically dispatches installation to the one or more install slave machines. The modification would be obvious because one of ordinary skill in the art would be motivated to provide redundant data backup or testing platforms for diagnosing and monitoring software installation/performance.

As per **Claim 20**, the rejection of **Claim 19** is incorporated; however, Breggin does not disclose:

- wherein the one or more install slave machines comprise multiple computers.

Kruger discloses:

- wherein the one or more install slave machines comprise multiple computers *(see Column 4: 1-5, "The master computer is any computer on which the computer software can be properly installed, and for which such installation will be used as a model for installation of the software on other computer systems.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kruger into the teaching of Breggin to include wherein the one or more install slave machines comprise multiple computers. The modification would be obvious because one of ordinary skill in the art would be motivated to properly install software programs in computer systems.

As per **Claim 22**, the rejection of **Claim 19** is incorporated; and Breggin further discloses:

- wherein the baseline-update interface comprises a web-based user interface *(see Column 10: 40-42; "In Web-based application, the installed database or file can be incorporated into one or more web pages.").*

However, Breggin does not disclose:

- allowing baseline updates across SKU, language, operating system, and custom/non-custom installs, in combination or all at once.

Official Notice is taken that it is old and well-known within the computing art to allow baseline updates across SKU, language, operating system, and custom/non-custom installs, in combination or all at once. A Web-based database management system typically allows a user to update various fields within a database. Therefore, it would have been obvious to one of ordinary

skill in the art at the time the invention was made to include allowing baseline updates across SKU, language, operating system, and custom/non-custom installs, in combination or all at once. The modification would be obvious because one of ordinary skill in the art would be motivated to allow a user to selectively update data.

As per **Claim 23**, the rejection of **Claim 19** is incorporated; and Breggin further discloses:

- wherein the attributes comprising modification date stamp information and file size information *(see Column 8: 24-29, "The database lists ... file size ('SIZE') ... last modification date of the file ('DATE') ...").*

However, Breggin does not disclose:

- wherein the attributes comprising security permissions information and checksum information.

Official Notice is taken that it is old and well-known within the computing art to define data in a database using various attributes. Data in a database often contains various attributes pertaining to the particular application of the data. For example, software installation data in a database often contains file permission information and file checksum information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the attributes comprising security permissions information and checksum information. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional useful data pertaining to the software installation to a user.

16. **Claim 21** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Breggin** in view of **Kruger** as applied to Claim 19 above, and further in view of US 2002/0156831 (hereinafter **Suorsa**).

As per **Claim 21**, the rejection of **Claim 19** is incorporated; however, Breggin and Kruger do not disclose:

- wherein the install controller communicates with the one or more install slave machines using Simple Object Access Protocol.

Suorsa discloses:

- wherein the install controller communicates with the one or more install slave machines using Simple Object Access Protocol *(see Paragraph [0052], "... messages that are exchanged between the gateway and the agents can be in the form of remote procedure calls that conform to the XML-RPC protocol, or the Simple Object Access Protocol (SOAP).")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Suorsa into the teaching of Breggin to include wherein the install controller communicates with the one or more install slave machines using Simple Object Access Protocol. The modification would be obvious because one of ordinary skill in the art would be motivated to provide a way to communicate between applications running on different operating systems with different technologies and programming languages.

Response to Arguments

17. Applicant's arguments with respect to Claim 1 have been considered, but are moot in view of the new ground(s) of rejection.

In the remarks, Applicant argues that:

a) Additionally, Kruger does not teach or suggest "creating installation data for a resource, based at least in part on the comparison, the resource including attributes **including a dynamic attribute and a static attribute**, the **dynamic attribute is an attribute that should have changed** between the previous software installation and the current software installation, the **static attribute is an attribute that should remain unchanged** between the previous software installation and the current software installation" (emphases added) as recited in claim 1. Although Kruger discloses comparing a computer system before and after the software installation, there is no teaching or suggestion by Kruger of creating installation data for a resource that includes both **dynamic and static attributes**. Contrary to the Office's contention, the cited portions of Kruger merely discloses creating a tree having nodes representing files; in fact, Kruger is silent on creating installation data for a resource that has a **dynamic attribute**, which is an attribute that **should have changed** between successive installations on the same target computer.

Examiner's response:

a) Examiner disagrees. Kruger discloses a dynamic attribute and a static attribute *(see Column 5: 18-29, "In both of the preceding embodiments, leaf nodes of the subtree or subtrees*

correspond to files, and contain information about the files in place of the files themselves. In one embodiment, such information is referred to as the node's properties and contains some or all of the file details ... These details may include the filename, last modification date, size, access permissions such as read only, and security information describing who is allowed access to the file and the type of access allowed."; Column 6: 51-58, "Registry file state retriever 225 reads the operating system registry file, such as the windows registry file in Microsoft Windows 95, and builds a subtree corresponding to the hierarchy of the registry file. For example, the Windows registry file arranges the keys and values in a hierarchical folder system and this hierarchy is used to build the subtree. Leaf nodes hold the values in the node's properties, and parents of these nodes store the keys in their properties.").

In the remarks, Applicant argues that:

b) Furthermore, Kruger does not teach or suggest "identifying from the installation data the **dynamic attribute that was not changed** in the current software installation" (emphasis added) as recited in claim 1. As noted above, Kruger is silent on creating installation data for a resource that has a dynamic attribute; thus, Kruger simply does not teach or suggest identifying the dynamic attribute that was not changed. Claim 1 requires the identification from the installation data attributes that **should have changed, but was not changed**, between successive installations on the same target computer. In sharp contrast, the cited portions of Kruger merely discloses that a difference calculator marks the terminal node in the tree it creates as the "same," which denotes that the state represented by the terminal node did not change when the new software was installed. Kruger simply does not contemplate any dynamic attributes (i.e.,

attributes that should have changed) in software installation, nor does Kruger identify **dynamic attributes that was not changed** for a given resource associated with a software installation.

Examiner's response:

b) Examiner disagrees. Kruger discloses identifying from the installation data the dynamic attribute that was not changed in the current software installation *(see Column 8: 34-40, "When difference calculator 234 compares a terminal node, the properties of the node are also compared, and if the properties of each corresponding node are the same, difference calculator 234 marks the terminal node in the tree it creates as the "same". This means the state represented by the terminal node did not change when the new software was installed.").*

In the remarks, Applicant argues that:

c) The Office acknowledges that Kruger does not disclose "presenting potential problems with the current software installation based on the identified dynamic attribute to facilitate verification of an installer for the software product development." (See 06/04/2007 Office Action at page 7.) The Office relies on Breggin for this subject matter and responds to Applicant's prior argument regarding Breggin by focusing on the prior reference to an alert or warning. (See 06/04/2007 Office Action at page 22.) However, attention is called to the fact that the crux of the argument does not hinge on the manner in which potential problems are presented, but rather the fact that the claim recites, "presenting potential problems with the current software installation based on the identified dynamic attribute to facilitate verification of an installer for the software product development." (Emphasis added.) Breggin explicitly states that, "An 'exception' is

typically a difference between corresponding fields in the installation and installed databases or files." (See Breggin at col. 9, lines 58-60.) Nothing in Breggin teaches or suggests presenting potential problems with a current software installation based on an identified dynamic attribute, where the dynamic attribute is an attribute that should have changed between the previous software installation .and the current software installation, but that did not change, as expected.

Examiner's response:

c) In response to Applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Kruger discloses the identified dynamic attribute (see Examiner's response (b) above). Breggin is relied upon for its specific teaching of presenting potential problems with the current installation. The combined teaching of Kruger and Breggin supports the conclusion that the claimed invention is directed to obvious subject matter.

Note that Applicant did not traverse the Examiner's assertion of Official Notice with regard to Claims 3, 7, 16, 22, and 23. Therefore, the "old and well-known within the computing art" statement is taken to be admitted prior art because Applicant has failed to traverse the Examiner's assertion of Official Notice (see MPEP § 2144.03).

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / RC
November 16, 2007

WEI ZHEN
SUPERVISORY PATENT EXAMINER